LISTING OF THE CLAIMS

1. (Original) A method for testing floating point hardware in a processor while executing a computer program, comprising:

executing a first set of code of said computer program without employing said floating point hardware, said first set of code having a first floating point instruction, thereby obtaining an emulated result;

executing said first floating point instruction utilizing said floating point hardware, thereby obtaining a hardware-generated result; and

comparing said emulated result with said hardware-generated result.

2. (Original) The method of claim 1 further comprising rendering said floating point hardware unavailable prior to said executing said first set of code of said computer program without employing said floating point hardware.

3. (Original) The method of claim 2 wherein said processor represents an Itanium™ processor, said rendering said floating point hardware unavailable including setting at least one of a DFH and a DFL bit in a processor status register of said processor.

4. (Original) The method of claim 2 wherein said processor represents a PA-RISC™ processor, said rendering said floating point hardware unavailable including clearing a CR10 co-processor control register of said processor.

5. (Original) The method of claim 2 wherein said rendering said floating point hardware unavailable includes writing a first predefined value into a register in said processor.

6. (Original) The method of claim 5 further comprising rendering said floating point hardware available for executing instructions of said computer program prior to said executing said first floating point instruction utilizing said floating point hardware.

7. (Original) The method of claim 6 wherein said rendering said hardware available includes writing a second predefined value into said register in said processor.

8. (Original) The method of claim 7 wherein said processor represents an Itanium™ processor, said rendering said floating point hardware available including clearing at least one of a DFH and a DFL bit in a processor status register of said processor.

9. (Original) The method of claim 7 wherein said processor represents a PA-RISC™ processor, said rendering said floating point hardware available including setting a CR10 co-processor control register of said processor.

10. (Original) The method of claim 1 wherein said obtaining said hardware-generated result includes obtaining a trap result after said first floating point instruction is executed utilizing said floating point hardware.

11. (Original) The method of claim 1 wherein said obtaining said emulated result includes obtaining a hardware-generated trap result after said processor encounters said first floating point instruction while said floating point hardware is unavailable.

12. (Original) The method of claim 1 wherein said computer program represents a field application program.

13. (Original) A method for detecting failure in floating point hardware of a processor while executing a computer program, comprising:

entering a diagnostic mode, including

executing a first floating point operation of said computer program by emulating said floating point operation with a set of non-floating point operations, thereby obtaining an emulated result,

executing said first floating point operation utilizing said floating point hardware, thereby obtaining a hardware-generated result, and

comparing said emulated result with said hardware-generated result to detect said failure to detect said failure;

determining whether diagnostic mode is to be continued; and

resuming execution of said computer program in a non-diagnostic mode if said diagnostic mode is to be discontinued, said non-diagnostic mode involving performing

floating point operations of said computer program without emulating with non-floating point operations.

14. (Original) The method of claim 13 said entering said diagnostic mode further comprising rendering said floating point hardware unavailable prior to said executing said first floating point operation by said emulating.

15. (Original) The method of claim 14 wherein said processor represents an Itanium™ processor, said rendering said floating point hardware unavailable including setting at least one of a DFH and a DFL bit in a processor status register of said processor.

16. (Original) The method of claim 14 wherein said processor represents a PA-RISC™ processor, said rendering said floating point hardware unavailable including clearing a CR10 co-processor control register of said processor.

17. (Original) The method of claim 14 wherein said rendering said floating point hardware unavailable includes writing a first predefined value into a register in said processor.

18. (Original) The method of claim 17 wherein said entering said diagnostic mode further including rendering said floating point hardware available for executing

instructions of said computer program prior to said executing said first floating point operation utilizing said floating point hardware.

19. (Original) The method of claim 18 wherein said rendering said hardware available includes writing a second predefined value into said register in said processor.

20. (Original) The method of claim 19 wherein said processor represents an Itanium™ processor, said rendering said floating point hardware available including clearing at least one of a DFH and a DFL bit in a processor status register of said processor.

21. (Original) The method of claim 19 wherein said processor represents a PA-RISC™ processor, said rendering said floating point hardware available including setting a CR10 co-processor control register of said processor.

22. (Original) The method of claim 13 wherein said obtaining said hardware-generated result includes obtaining a trap after said first floating point operation is executed utilizing said floating point hardware.

23. (Original) An article of manufacture comprising a program storage medium having computer readable code embodied therein, said computer readable code being configured to test floating point hardware in a processor while executing a computer program, comprising:

computer readable code for executing a first set of code of said computer program without employing said floating point hardware, said first set of code having a first floating point instruction, thereby obtaining an emulated result;

computer readable code for executing said first floating point instruction utilizing said floating point hardware, thereby obtaining a hardware-generated result; and

computer readable code for comparing said emulated result with said hardware-generated result.

24. (Original) The article of manufacture of claim 23 further comprising computer readable code for rendering said floating point hardware unavailable prior to said executing said first set of code of said computer program without employing said floating point hardware.

25. (Original) The article of manufacture of claim 24 wherein said computer readable code for rendering said floating point hardware unavailable includes computer readable code for writing a first predefined value into a register in said processor.

26. (Original) The article of manufacture of claim 25 further comprising computer readable code for rendering said floating point hardware available for executing instructions of said computer program prior to said executing said first floating point instruction utilizing said floating point hardware.

27. (Original) The article of manufacture of claim 26 wherein said computer readable code for rendering said hardware available includes computer readable code for writing a second predefined value into said register in said processor.